

Systems [3.0]

1. [2.0 points] When defining a complex system using a systemic approach, one principle is that a system can be broken into smaller subsystems.

Programming languages offer functions, classes and modules, but operating systems also offer their own mechanisms.

- Present two mechanism offered by UNIX that allow the programmer to break a big complex system into smaller subsystems.
- Present two differences between such mechanisms

2. [1.0 points]

- Describe what is the objective of the use of Interaction Diagrams

Requirements definition [4.0]

3. [2.0 points] Requirement Definition is a fundamental step in the development of a systems.

- Present the various steps of a Software development Lifecycle.
- Describe the reason for the existence and position of the Requirements Definition step in the Software Development Life cycle.
- Describe what is the result in this step and on what other stages these results are used.

4. [2.0 points] The following sentence is a example of a bad requirement:
The system should provide fast response to a lot of clients.

- Describe the main problem with this requirements
- Produce two possible requirements that could replace the example.

Concurrency [4.0]

5. [2.0 points] Concurrency and parallelisms are two possible characteristics of a system.

- What is concurrency?
- What is parallelism?
- Describe a system that offers concurrency without parallelism. Justify.
- Describe the advantages of such a system having concurrency without parallelism.

6. [2.0 points] The scheduler is one component of the modern operating systems.

- What is the role of the scheduler on a Operating System?
- Describe two differences between preemptive and non-preemptive scheduler.

Processes / Threads / Synchronization [4.0]

7. [2.0 point] The following code creates three threads:

```
for (int i = 0; i < 3; i++){  
    pthread_create(&t_ids[i], NULL, thread_code, i);  
}
```

Each thread creates a string that should be processed after the thread termination.

- Write the code inside the thread that allows it to return a string.
- Write the code that allows the access to the three returned string after the threads terminate.

8. [2.0 point]

- Explain what can happen when a thread that is holding a lock (inside a critical region) tries to lock the same mutex.

Interprocess communication (IPC) [5.0]

9. [2.0 point] Suppose that a system is composed of two threads that need to exchange data between them.

Suppose that each exchanged message is of a defined C structure (for instance **typedef struct {...} msg_type**).

- What IPC mechanism (communication channel) should be used to exchange data between such threads.
- Compare the proposed mechanism another one by presenting one advantage of the proposed mechanism .
- Write the code that creates such communication mechanism: Declare and initialize the necessary variables.

10. [1.0 point] UNIX also offers the possibility of using shared memory between different processes and threads.

- Describe how shared memory can be used in various threads.
- Present one advantage of using shared memory over a regular communication channel.
- Present one disadvantage of using shared memory over an regular communication channel.

11. [2.0 point] In the project the Applications/KVS-Library and the KVS-localServer used Unix domain sockets to communicate.

- Explain the changes necessary to make on the code so that the applications and KVS-localserver could be deployed in different machines.